DEEP Projects

WP3: System Architecture

H.-C. Hoppe, A. Atanasov

Outline

WP3 Activities since last F2F (October 2018)

- Formalities (Tasks, Deliverables)
- Re-architecting the ESB
- Network federation
- SDVs and evaluators
- MSA test & evaluation codes

Intel technology update

- Optane DC Persistent Memory (aka non-volatile DIMMs)
- AI support in Cascade Lake CPU
- Programming the Stratix FPGA
- “Intel One API”

WP3 Update

Hans-Christian Hoppe

WP3 Objectives

No	Description	Deliverables
1	Define the system architecture and create the high-level specification for the DEEP-EST prototype in close co-design collaboration.	D3.1, D3.2, D3.2U ✓
2	Manage technical project risks via a sequence of evaluation platforms, and provide early access to new technology for SW developers.	
3	Assess fulfilment of the high-level system and module specifications in light of WP4 results.	D3.3, D3.4
4	Report on achieved results and lessons learnt and propose an evolution of the MSA towards Exascale.	D3.5

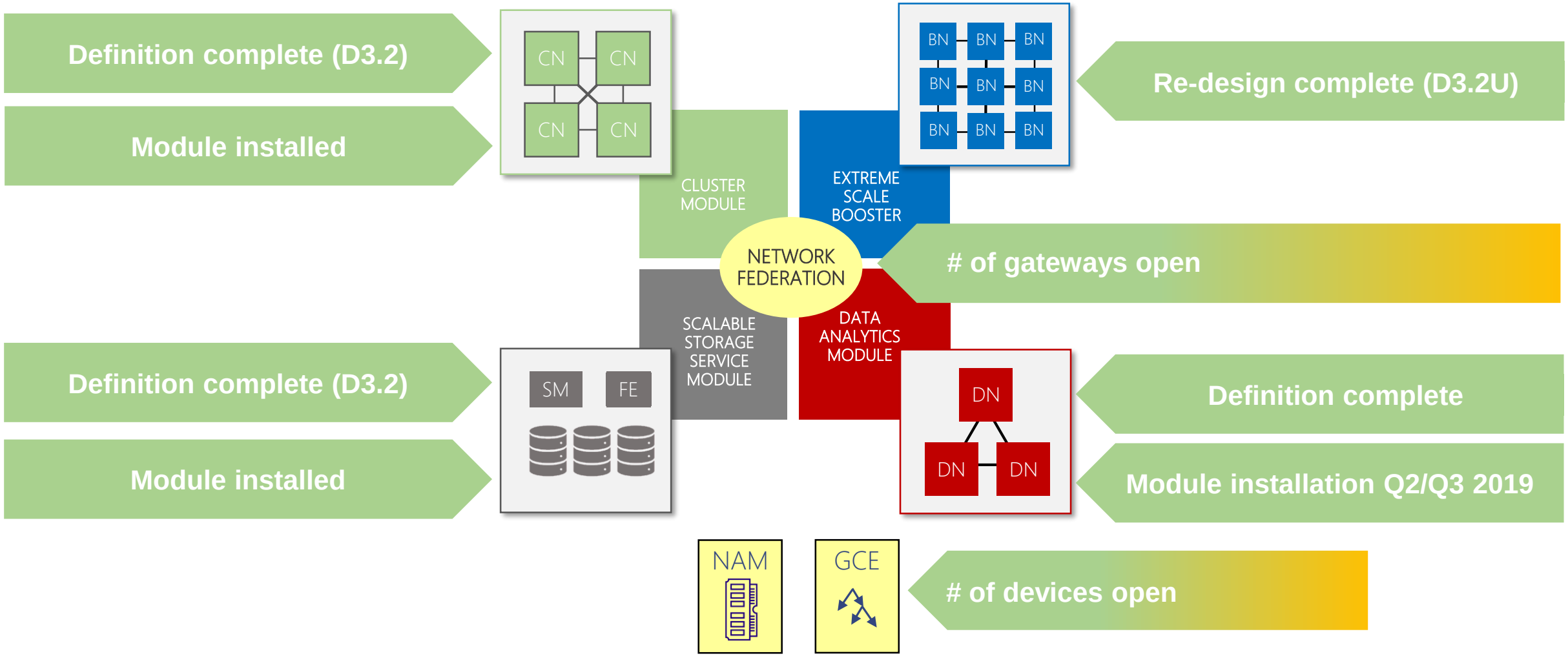
WP3 Structure & Status

JUELICH	Intel	BADW-LRZ	Megware	UHEI	EXTOLL
20 PM	26 PM	3 PM	14 PM	6 PM	6PM

Task	Description	Partner	Term	Status
3.1	System Architecture in Co-Design	Jülich	M1-M18	✓
3.2	Evaluators and SDVs	Intel	M3-M30	Ongoing
3.3	System Assessment	Intel	M18-M33	Ongoing
3.4	Architecture evaluation and outlook	Intel	M36-M45	Not Started

Deliverable	Description	Partner	Delivery	Status
D3.1	System Architecture	Jülich	M6	✓
D3.2	High Level System Design	Jülich	M12	✓
D3.2U	High Level System Design (Update)	Jülich	M19	✓
D3.3	Tests for Prototype Assessment	Intel	M24	Ongoing
D3.4	Prototype Assessment	Intel	M33	Not Started
D3.5	Modular Supercomputer Architecture Assessment & Outlook	Intel	M45	Not Started

Modular Supercomputing Architecture & Design



ESB – New Architecture & Design

Result of M12 external review

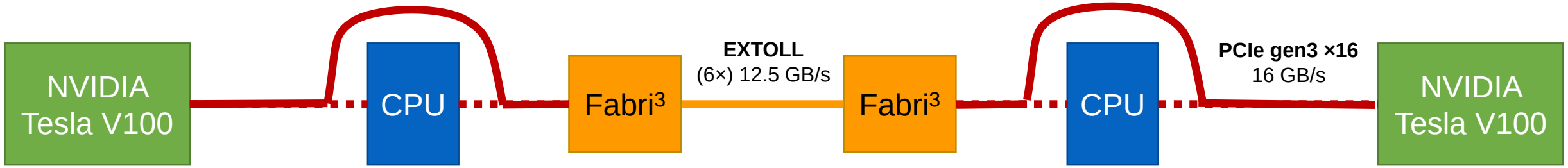
- MUST have a different, many-core architecture in the ESB
- Strong hint to use the NVIDIA V100 GPGPU
- Request to propose new architecture until M15 & go through review in M18/M19

Re-architecturing principles

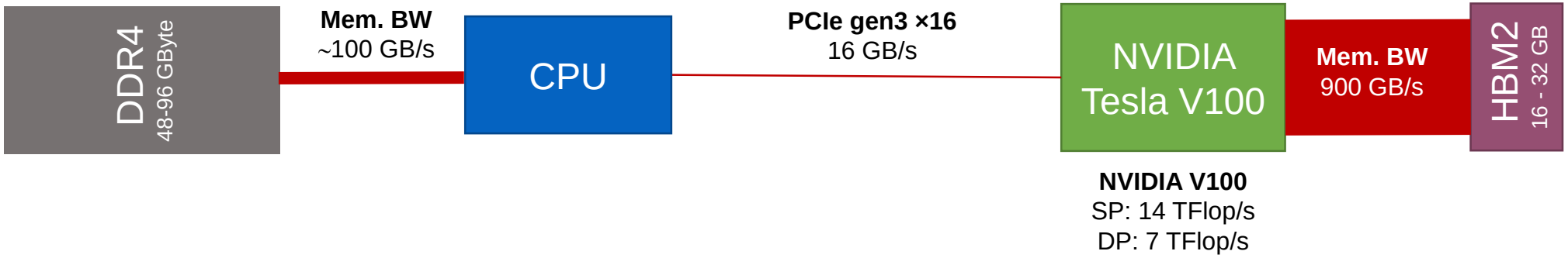
- Back to DEEP Booster philosophy – highly tuned many-core/throughput processor with minimal overhead for “hosting”, highest scalability
- Match compute capability with achievable fabric bandwidth (100 Gb/s)
- Avoid introducing complexity
- Provide end-to-end MPI communication capability
- Streamline integration with EXTOLL fabric
- Involve application partners

New ESB Node

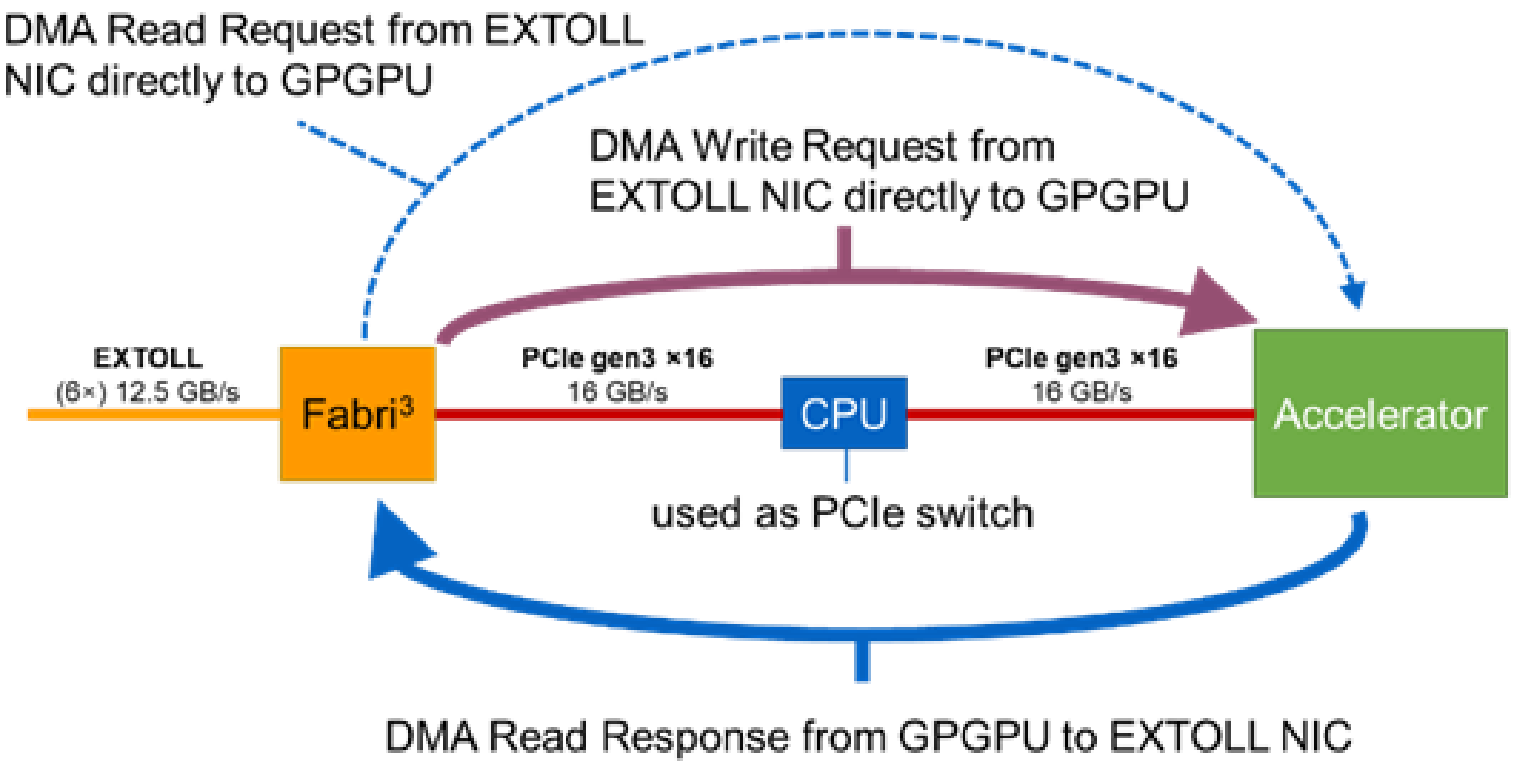
Balanced end-to-end GPGPU ↔ GPGPU communication



BUT Unbalanced GPGPU vs. CPU memory bandwidth



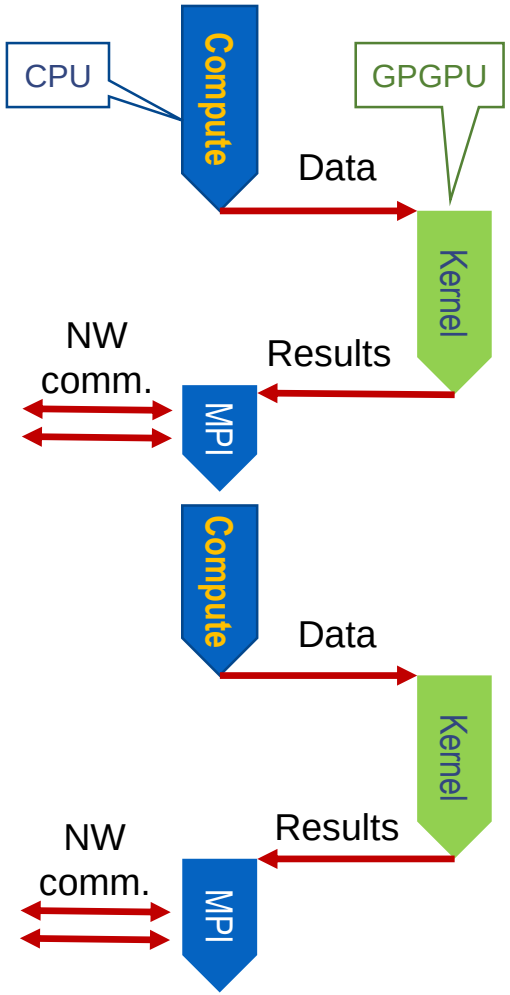
EXTOLL Support for GPUDirect RDMA



GPUDirect RDMA Enables full use of PCI gen3 bandwidth between GPGPU and EXTOLL fabric, minimizing CPU involvement

ESB Programming Principles

Conventional CPU/GPGPU Offload



Strictly **reduce calculation** on the **host**

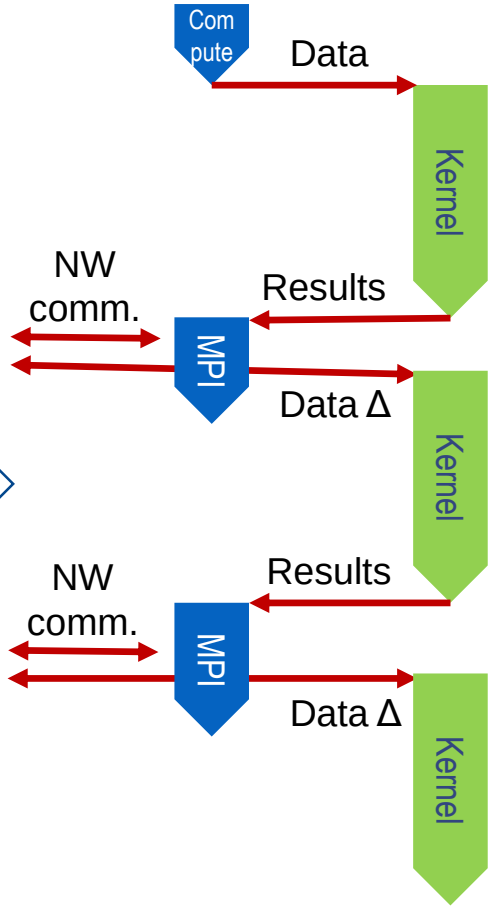
Make all **data fit** into **GPGPU memory**

Reduce Host ↔ GPGPU communication to the MIN

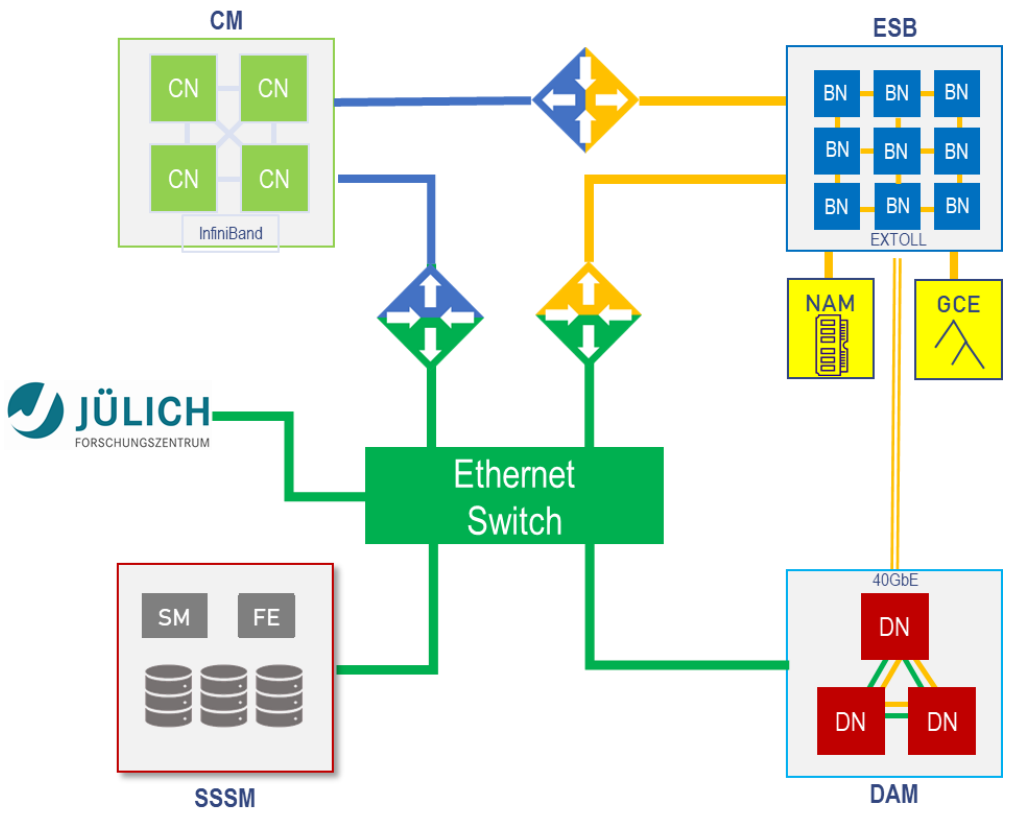
System SW & programming model

Enable **direct transfer** of **MPI data to GPGPU**

Optimized ESB CPU/GPGPU Offload



Network Federation



IB ↔ EXTOLL
- Carries IP and MPI
- Developed in T5.3

IB ↔ Ethernet
- Linux functionality
- Images available

EXTOLL ↔ Ethernet
- Linux functionality

Tested & benchmarked (details see WP5)

Important decision to be taken:
# of gateways of each sort

Need to have a close look at the GROMACS results on Jureca!

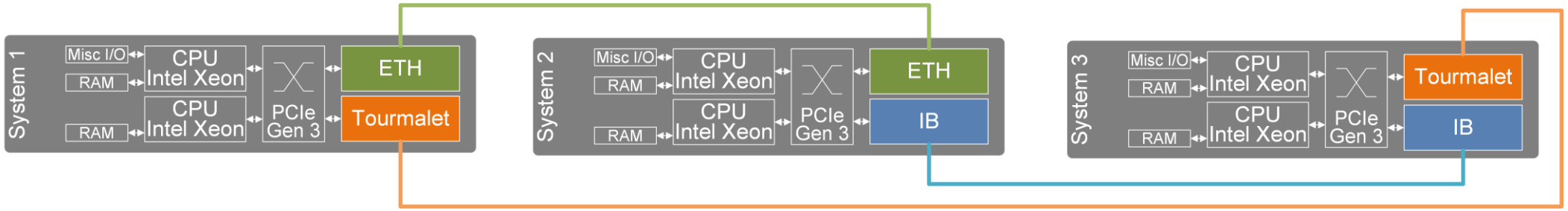
SDVs and Evaluators

Intel® Arria® 10 FGPA SDVs

- 2 systems with Skylake CPU and Arria 10 FPGA have been available at Jülich for a while

Network gateway evaluators

- Three DP Skylake servers with Infiniband, Ethernet & EXTOLL NICs
- Can be used to configure producer → gateway → consumer chains for each gateway version



EXTOLL ESB integration evaluators

- Two DP Xeon systems with EXTOLL NIC and V100 installed
- Serve to test and optimize EXTOLL driver integration with GPUDirect RMA and end-to-end ESB communication performance

Sun-setting of DEEP-ER SDV KNL nodes

Test & Evaluation Codes – T3.3 & D3.3

Main objective of T3.3

"Assess fulfilment of the high-level system and module specifications in light of WP4 results."

Approach taken

- Identify "simple" test and evaluation codes, working together with WP2
- These test *critical* performance parameters
  - *Of each module – like CPU & accelerator compute performance, intra-node communication, communication across the fabric*
  - *Of the system – focusing on inter-module communication*
- "Simple" meaning that codes are available plus easily adapted, compiled and run
- As far as possible, use commonly accepted benchmarks or micro/proto-applications

Modules: What to Test & Measure?

Compute performance*

- CPU and GPGPU: delivered (FP) performance for compute-bound problems, using all available parallelism → Linpack, miniGhost, LCALS (souped-up Livermore loops), …
- FPGA: not clear what to pick – maybe best to start with Astron application, & add a meaningful DL inference benchmark

Memory performance* (DRAM, HBM and NVRAM)

- CPU and GPGPU: STREAMS for bandwidth, HPC Challenge Random Access for latency, HPCG, …
- FPGA: TBD

Communication performance intra-node*:

- CPU+GPGPU (DAM/ESB): measure data transfer latency & BW between CPU and GPGPU memory across PCI Express (NVIDIA bandwidth test, STREAMS between CPU and GPGPU memory, Random Access?)
- CPU+FPGA: ditto, using OpenCL

*: energy will be measured in each case, as far as platforms allow

Modules: What to Test & Measure?

Communication performance inter-node*:

- MPI performance (CM, DAM): MPI Linktest (T2.1) and IMB (collectives, bisection)
  - *ESB: modify MPI Linktest and IMB where necessary*
- ESB raw bandwidth over EXTOLL: adapt STREAMS and maybe Random Access to work across memory of several ESB nodes
- I/O bandwidth to NAM: IOR

Combined performance*:

- CM, DAM and ESB: HPL, HPCG, Random Access on module, Graph500?

*: energy will be measured in each case, as far as platforms allow

MSA System: What to Test & Measure?

Communication performance inter-module*:

- MPI performance: MPI Linktest (T2.1) and IMB (collectives, bisection) across module pairs
- I/O bandwidth to SSSD: IOR from modules to SSSM
- Do we need IP performance measurements?

*: energy will be measured in each case, as far as platforms allow

Next Steps

- Agree on list of benchmark/evaluation codes with WP4 and WP2
- Get DDG to ratify the evaluation code list
- Write up D3.3 and pass into internal review
- Put the evaluation codes under JUBE control